Privacy through Fake yet Semantically Real Traces

Vincent Bindschaedler
UIUC
bindsch2@illinois.edu

Reza Shokri
UT Austin
shokri@cs.utexas.edu

ABSTRACT

Camouflaging data by generating fake information is a well-known obfuscation technique for protecting data privacy. The effectiveness of this technique in protecting users' privacy highly depends on the resemblance of fake information to reality, such that an adversary cannot easily filter such fake information out. In this paper, we focus on a very sensitive and increasingly exposed type of data: location data. There are two main scenarios in which fake traces are of extreme value to preserve location privacy: publishing datasets of location trajectories, and using location-based services. Despite advances in protecting (location) data privacy, there is no quantitative method to *evaluate* how realistic a synthetic trace is, and how much utility and privacy it provides in each scenario. Also, the lack of a methodology to *generate* privacy-preserving fake traces is evident. In this paper, we fill this gap and propose the first statistical metric and model to generate fake location traces such that both the utility of data and the privacy of users are preserved.

We build upon the fact that, although geographically they visit distinct locations, people have strongly semantically similar mobility patterns, for example, their transition pattern across activities (e.g., working, driving, staying at home) is similar. We define a statistical metric and propose an algorithm that automatically discovers the hidden semantic similarities between locations from a bag of real location traces as seeds, without requiring any initial semantic annotations. We guarantee that fake traces are geographically dissimilar to their seeds, so they do not leak sensitive location information. We also protect contributors to seed traces against membership attacks. Interleaving fake traces with mobile users' traces is a prominent location privacy defense mechanism. We quantitatively show the effectiveness of our methodology in protecting against localization inference attacks while preserving utility of sharing/publishing traces.

1. INTRODUCTION

Fake (dummy) information can protect privacy and security in many different systems such as web search [11], anonymous communications [3, 8], authentication systems [12], and statistical analysis [17, 23]. In all these scenarios, the main challenge and the open problem is to generate context-dependent fake information that resembles genuine user-produced data and also provides an acceptable level of utility while enhancing privacy of users.

In this paper, we propose a systematic approach for preserving privacy of *location* data using fake traces. We focus on two practical scenarios: sharing location with location-based services, and publishing location datasets e.g., for research. In location-based systems, users hide their true location among fake location traces while connecting to a server to obtain contextual information about their whereabouts. This protects them against the location inference attacks. The benefit of the fake injection approach with respect to other obfuscation techniques, such as location perturbation [1, 10, 28], is that it does not reduce the users' experienced service quality. Users only incur the overhead of filtering out the received information about fake locations. In publishing privacy-preserving location datasets, the purpose is to preserve the general statistics about human mobility. There is a utility loss associated with fake traces as they might not fully preserve all the characteristics of real traces. The challenge is to generate synthetic traces that semantically resemble the real traces yet do not leak information about the exact geographic locations visited by any particular individual. This gives rise to a tradeoff between utility and privacy that is inherent to privacy-preserving systems.

There has been some preliminary work on using fake location queries to protect users' location privacy [6, 13, 15, 26, 32]. However, they are based on very simple heuristics such as i.i.d. location sampling, sampling locations from a random walk on a grid with uniform probability, and using road trip algorithms to generate driving traces between two random locations. In this paper, we quantitatively show that these methods fail to protect location privacy against inference attacks. Besides, what these methods are missing are (i) a *metric* that captures how realistic a synthetic location trace is with respect to human mobility, so it cannot be easily detected by attacker, (ii) a *generative model* that produces samples of synthetic yet realistic traces according to such a metric, while preserving utility and ensuring that the synthetic traces do not themselves leak information about any individual. In this paper, we present the first formal methodology to solve these problems and to generate fake yet semantically real location traces for protecting location privacy. We also enforce, and quantitatively measure, privacy against location privacy attacks.

Our scheme is based on the fact that mobility patterns of different individuals share some semantic features, regardless of which geographic locations they visit. These common features of human mobility stem from their similar lifestyles. The mobility patterns share a similar structure that reflects the general behavior of a population (even at a high level [29]). We model the mobility of each individual in two dimensions: *geographic* and *semantic*. The geographic features are mostly specific to each individual (hence are sensitive),

whereas the semantic features are mostly generic and representative of human mobility behavior (hence are useful). We extract the common semantic features of mobility patterns and use them to generate realistic synthetic traces, without leaking the geographic features of any individual's locations.

Consequently, we define two metrics to quantify the similarity between human mobility models: The geographic similarity metric between two individuals captures how correctly we can predict locations of one knowing the mobility model of the other one. This metric helps us to capture the spatiotemporal information leakage of a fake trace about a real trace. The semantic similarity metric reflects how well two traces match in terms of their semantic features. We assume that we have a dataset of real traces. We develop an algorithm that automatically learns the semantic correlation between locations and transitions between locations.[1] To generate semantically realistic fake traces, we transform a (real) seed trace into the semantic domain and probabilistically sample (fake) location traces that are consistent with it. Thus, a generated sample resembles a typical sequence of locations that could have been visited by some real individual. We then design a *rejection sampling* assertion to ensure that a fake trace does not leak information about the locations in its seed trace, i.e., we reject those who do not meet the privacy requirements. Thereby, we protect privacy of seed traces against the following threats: *Inference* attacks (to learn which locations the seed contributors have visited), and *membership inclusion* attack (to learn if a particular individual with certain semantic habits has been in the seed dataset). If a fake trace's geographic similarity or its intersection with its real seed trace exceeds a given threshold, we reject it and sample a new trace. We also ensure that the semantic similarity between fake and seed traces cannot be used against anonymity of seed traces. To this end, we reject the fake trace if there is no $k-1$ alternative real traces that could have been the seed for generating the fake (i.e., the differential semantic similarity with the fake trace is below a threshold). This additionally guarantees a *plausible deniability* for each seed trace. The resulting pool of fake traces can later be drawn upon for use by e.g., users' smartphones. The fake traces generated from the seed database can be used to protect location privacy of any user (not the contributed of the seed database). The privacy tests guarantee that we preserve privacy of seed traces. Additionally, we show that the generated fakes can significantly protect privacy of LBS users against inference attacks.

Our Contributions: In summary, the novelty of this work is twofold. (1) We introduce the notion of semantic similarity for mobility patterns, we propose both a metric for it, and an algorithm to quantify the semantic similarity between location traces. We also automatically learn the semantic relation between locations. (2) We propose a generative model for fake location traces that are semantically similar to real traces. We also guarantee plausible deniability to individuals whose real traces are used as seed in our algorithm. Our software tool, given a set of real location traces, generates fake traces based on our theoretical framework. We run our algorithms on a real-world dataset collected by Nokia [14]. We show the effectiveness of our fake traces in protecting privacy of users in two main scenarios: location-based services and published location datasets, while preserving utility.

[1]Note that we do not annotate the locations (as home, work, ...), nor we use any annotation as an input to compute the similarity between locations. We only rely on location traces as input.

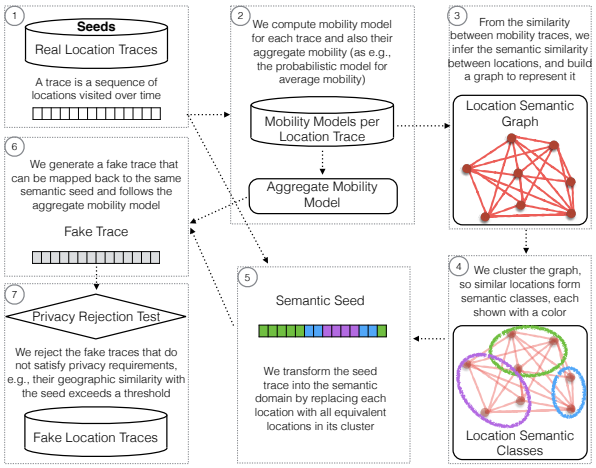


Figure 1: Sketch of the proposed scheme.

2. OUR SCHEME

In this section, we present a sketch of, and describe the main intuition behind, our scheme for generating fake traces. We assume that time and space are discrete, so a location trace is represented as a sequence of visited locations over time. In our scheme, we generate a fake trace through a multiple step process. Figure 1 illustrates the details.

2.1 Computing the Semantic Similarity

The first step is to compute the semantic similarity between the set of locations in an area. We learn these similarity values automatically from a *dataset* of real location traces. For each trace (i.e., the sequence of locations visited by a single individual) in the dataset, we first compute a probabilistic *mobility model* that represents the visiting probability to each location and transition probability among the locations (see Section 3.1). The mobility model encompasses the spatiotemporal behavior of each individual with respect to different locations. Time, duration, and probability (frequency) of visiting a location, as well as the probable comings from and goings to locations are all computable from the mobility model. So, it implicitly reflects the types of activities that an individual might carry out in each location (and over a sequence of locations).

We analyze and discover the semantic relation between different locations in a consistent manner by considering all locations together. To this end, we propose a *semantic similarity metric* (see Section 3.4). Intuitively, we assign a higher similarity value to the pair of locations at which different individuals have similar spatiotemporal activities. Thus, our metric tries to find the optimal way to map the visited locations in a pair of traces such that the mapping maximizes the statistical similarity between their mobility models. The semantic similarity metric is therefore the statistical similarity between mobility models under the optimal mapping between locations. This means that if we were to translate the locations of this pair according to the discovered best mapping, they would follow the same mobility model when their semantic similarity is high. For example, consider Alice and

Bob spending all day at their respective work locations w_A and w_B, and all night at their respective home locations h_A and h_B. Obviously, their mobility models are semantically very similar, although it might be the case that $h_A \neq h_B$ and $w_A \neq w_B$. In this example, the best semantic mapping between locations will be $w_A \leftrightarrow w_B$ and $h_A \leftrightarrow h_B$.

For each pair of mobility models for traces in our dataset, we compute their semantic similarity as well as the best semantic mapping between their locations. We then aggregate all the location matchings across all trace pairs, with weights based on the semantic similarity between mobility models, and construct a *location semantic graph*, where the nodes are locations and the weight of the edges is the average semantic similarity between the locations over the dataset.

2.2 Forming Location Semantic Classes

The location semantic graph enables us to find what locations have similar meanings for different people, so they have similar activities in those places. The locations that have higher semantic similarity can be grouped together to represent one *location semantic class*. To this end, we run a clustering algorithm on the location semantic graph to partition locations into distinct classes. Locations that fall into the same class are visited in the same way by different people regardless of their geographic positions.[2] Thus, we can consider them as being semantically equivalent. So, using the notations of our previous example, w_A and w_B should belong to the same cluster that can represent "workplace" locations, and h_A and h_B should be grouped into another cluster representing residential or "home" locations.

2.3 Generating a Fake Trace

We use the location semantic classes as the basis to generate fake traces. In addition to being semantically realistic, the fake traces must be geographically consistent with the general mobility of individuals in the considered area. For example, the speed of moving in some locations differs depending on the time, or the probabilities of taking different paths is different. To capture these patterns, we compute an *aggregate mobility* model from the traces in our dataset. We can, for example, compute this by averaging the mobility models that we constructed on the traces.

The goal is to generate fake traces that are semantically similar to real traces. In order to construct a fake trace, our algorithm starts with a *seed* trace and converts it to a probabilistically generated semantically similar synthetic trace which is consistent with the aggregate mobility model. We pick the seed trace at random from the trace dataset. The seed trace, similar to other location traces in the database, is composed of a sequence of geographic locations. In our algorithm, we first *transform* the geographic seed trace into the semantic domain, then we use the transformed semantic trace to *sample* from the state space of all geographic traces that could have been transformed to the same semantic trace. The transformation and sampling procedures, which are at the heart of this step, are done as follows.

For the seed trace *transformation*, we replace the geographic locations in the seed with the locations that are in the same semantic class. This seed *semantic trace* is a sequence of location sets. For the fake trace *sampling*, we address the following problem. We want to construct a trace that follows the aggregate mobility model under the constraint that its locations over time are subset of locations of the seed semantic trace. Hence, both the fake trace and the seed trace can be transformed to the same semantic trace. We add some randomness to the locations in the semantic trace to allow higher number of possible fake traces. Many algorithms can be used to sample the fake trace that satisfies our constraints. We make use of dynamic programming algorithms that construct the traces efficiently (Section 4).

We can repeatedly generate fake traces from each seed trace in the dataset, each of which having a probability according to the aggregate mobility model. After generating each fake trace, however, we need to make sure that it is not geographically similar to the seed trace. This is because we do not want to leak information about the real seed trace. To this end, we add a test to compute the *geographic similarity* (see Section 3.3) between the seed trace and the fake trace to *reject* the sample traces that are more similar than a threshold to the seed trace. Thus, we make sure that the semantically similar synthetic traces are indeed geographically dissimilar to the traces in our dataset, hence not leaking information about visited locations in the real traces.

[2]Their visit probability, time of visit, and the probabilities of transition from/to them to/from other locations of the same type is similar.

$\mathcal{R}$	Set of locations
R	Number of locations
r	A location
$\mathbf{r}$	Random variable associated with a location
T	Number of time periods
τ	A time period
$p(u)$	Transition probability matrix of user u
$\pi(u)$	Visiting probability vector of user u
$\langle p(u), \pi(u) \rangle$	Mobility profile of user u
$p^{\mathbf{x}}_{\mathbf{y}}(u)$	Probability of $\mathbf{x}$ given $\mathbf{y}$ according to u's mobility model
$d(\cdot)$	A distance function (between locations)
$M_d(p, q)$	Mallows distance between probability distributions p and q based on a distance function $d(\cdot)$
σ	A permutation function
$\mathsf{sim}_{\mathsf{G}}(u, v)$	Geographic similarity between mobility of u and v
$\mathsf{sim}_{\mathsf{S}}(u, v)$	Semantic similarity between mobility of u and v
σ^v_u	Optimal semantic mapping between locations of u and v
$\mathcal{S}$	Set of real traces used as seeds to generate fake traces
$\langle \bar{p}, \bar{\pi} \rangle$	Aggregate mobility model
$\mathcal{C}$	A partition on $\mathcal{R}$, representing location semantic classes. $\mathcal{C}_i$ is the set of locations in class (partition) i
$\mathcal{F}$	A set of fake locations generated from $\mathcal{S}$

Table 1: Table of notations

3. MOBILITY SIMILARITY METRICS

In this section, we present a probabilistic model for mobility, and propose two metrics to analyze the geographic and semantic similarity between two mobility models. Table 1 presents the list of notations that we use in this paper.

3.1 Mobility Model

We model the user mobility as a time-dependent first-order Markov chain on the set of regions (locations). As users have different behavior and mobility patterns during different periods of time, we assume that time is partitioned into time periods, e.g., morning - afternoon - evening - night. So, the mobility profile $\langle p(u), \pi(u) \rangle$ of a given user u is a *transition* probability matrix of the Markov chain associated

with the user's mobility (from a region to another), and the user's *visiting* probability distribution over the regions, respectively. Note that these probabilities are dependent on each other, and together they constitute the joint probability of two regions that are subsequently visited by the user. The entry $p^{r'}_{r,\tau,\tau'}(u)$ of $p(u)$ is the probability that user u will move to region r' in the next time instant (which will be in time period τ'), given that she is now (in time period τ) in region r. The entry $\pi^r_\tau(u)$ is the probability that user u is in region r in time period τ. Let the random variable $\mathbf{A}^t_u$ represent the actual location of user u at time t, and τ^t be the time period associated with $\mathbf{A}^t_u$. So, the mobility profile of a given user u consists of the following probabilities:

$$p^{r'}_{r,\tau,\tau'}(u) = \mathbb{P}\mathrm{r}\{\mathbf{A}^{t+1}_u = r' \mid \mathbf{A}^t_u = r; \tau^{t+1} = \tau', \tau^t = \tau\},$$
$$\pi^r_\tau(u) = \mathbb{P}\mathrm{r}\{\mathbf{A}^t_u = r; \tau^t = \tau\} \quad (1)$$

This Markovian model can predict the location of an individual to a great extent, as it takes both location and time aspect into account. It can become even more precise, by increasing its order, or by enriching its state. We can, for example, include multiple granularities of locations, and model the mobility of a user on the set of e.g., pair of (location, neighborhood) in addition to the time dimension. Our framework can incorporate all these new dimensions similar to the way we model the time periods. To learn the probabilities of the mobility profile (1), from location traces, we can use maximum likelihood estimation (if the traces are complete) or make use of algorithms such as Gibbs sampling (if the traces have missing locations or are noisy) [27].

3.2 Mobility Similarity Metrics

We propose two metrics to compare the mobility of two users and compute their similarities: *geographic* and *semantic* similarity. In this subsection, we describe the intuition behind these metrics, and in the following subsections, we formally define and provide the algorithms to compute them.

The *geographic similarity* metric captures the correlation between location traces that are generated by two mobility profiles. It reflects if two users visit similar locations over time with similar probabilities and if they move between those locations also with similar probabilities. Using this metric, for example, two individuals who live in the same region A and their workplace is in the same region B potentially have very similar mobilities, as they spend their work hours in B and most of their free time in A.

There are very few people whose mobility patterns have a high geographic similarity with each other. However, if we ignore the exact locations that are visited by different people, we observe that they share similar patterns for visiting locations with similar semantics (locations therein they have similar activities). For example, most people visit and stay at a single location from each evening until its subsequent morning. These locations differ from one individual to another, but have the same semantic for them: home.

One can imagine the semantic dimension of locations as a coloring on the locations in a map. Instead of computing the correlation between location traces at the geographic level, we can also compute such correlation at the semantic level (by reducing the set of locations to the set of colors and computing the similarity of color traces). This is the intuition behind our *semantic similarity* metric. In this case, if the pair of locations that two individuals visit over time have the same semantic, their mobility models are also semantically similar (even if they are in two different cities, i.e., have no geographic similarity). Hence, in this example, if we transform trace A by replacing its locations with their corresponding semantically similar locations in trace B, the transformed trace becomes geographically similar to B. So, two traces are semantically similar if their locations can be mapped to each other that accordingly one trace can be transformed to a geographically similar trace to the other.

3.3 Geographic Similarity Metric

We define this similarity metric based on the Earth Mover's Distance (EMD) for probability distributions. The EMD is widely used in a range of applications including image processing [24, 25]. The EMD can be understood by thinking of the two distributions as piles of dirt. In this interpretation, the EMD represents the minimum amount of work needed to turn one pile of dirt (i.e., one distribution) into the other; the cost of moving dirt being proportional to both the amount of dirt and the distance to the destination. The special case of EMD for probability distributions has been shown to be equivalent to the Mallows distance [16].

Let $\mathbf{X}$ and $\mathbf{Y}$ be discrete random variables with probability distributions p and q, such that $\mathbb{P}\mathrm{r}\{\mathbf{X} = x_i\} = p_i$ and $\mathbb{P}\mathrm{r}\{\mathbf{Y} = y_i\} = q_i$, respectively, for $i = 1, 2, \ldots, n$. We also have $\sum_i p_i = 1$ and $\sum_i q_i = 1$.

Definition 1. *(From [16]) Let $d(\cdot)$ be an arbitrary distance function between $\mathbf{X}$ and $\mathbf{Y}$. The* Mallows distance *$M_d(p, q)$ is defined as the minimum expected distance between $\mathbf{X}$ and $\mathbf{Y}$ with respect to $d(\cdot)$ and to any joint distribution function f for $(\mathbf{X}, \mathbf{Y})$ such that p and q are the marginal distributions of $\mathbf{X}$ and $\mathbf{Y}$, respectively.*

$$M_d(p, q) := \min_f \{\mathbb{E}_f\{d(\mathbf{X}, \mathbf{Y})\} : (\mathbf{X}, \mathbf{Y}) \sim f, \mathbf{X} \sim p, \mathbf{Y} \sim q\}, \quad (2)$$

where the expectation, minimized under f, is

$$\mathbb{E}_f\{d(X, Y)\} = \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij}\ d(x_i, y_j). \quad (3)$$

In addition to the constraints $\sum_{i=1}^n \sum_{j=1}^n f_{ij} = 1$ and $f_{ij} \geq 0$, for all i, j, the joint probability distribution function f must also satisfy $\sum_{i=1}^n f_{ij} = q_j$ and $\sum_{j=1}^n f_{ij} = p_i$.

Note that, for given p and q, the minimum f is easily computed by expressing the optimization problem as a linear program.

Using the previous definition, we define the geographic similarity metric based on the Mallows distance.

Definition 2. *Let $d(\cdot)$ be an arbitrary distance function. The* dissimilarity *between two mobility profiles $\langle p(u), \pi(u)\rangle$ and $\langle p(v), \pi(v)\rangle$ (belonging to individuals u and v), is defined as the expected Mallows distance of the next random locations $\mathbf{r}'$ and $\mathbf{r}''$ according to the mobility profiles of u and v, respectively. More formally, it is*

$$\mathbb{E}_{(u)}\{M_d(p^{\mathbf{r}'}_{\mathbf{r},\tau,\tau'}(u), p^{\mathbf{r}''}_{\mathbf{r},\tau,\tau'}(v))\}, \quad (4)$$

where $p^{\mathbf{r}'}_{r,\tau,\tau'}(u)$ and $p^{\mathbf{r}''}_{r,\tau,\tau'}(v)$ denote the conditional probability distributions of the next location, given the current location and the current and next time periods. The Mallows function is computed over random variables $\mathbf{r}'$ and $\mathbf{r}''$, and the expectation is computed over random variable $\mathbf{r}$ and time periods τ and τ'.

We define the geographic similarity *between mobility patterns of u and v as*

$$\mathsf{sim}_{\mathsf{G}}(u,v) := 1 - \mathbb{E}\{M_d(p^{\mathbf{r}'}_{\mathbf{r},\tau,\tau'}(u), p^{\mathbf{r}''}_{\mathbf{r},\tau,\tau'}(v))\}. \quad (5)$$

We compute the geographic *dissimilarity* using the law of total expectation. This also clarifies its meaning by showing more directly the role of the random variables.

$$\mathbb{E}\{M_d(p^{\mathbf{r}'}_{\mathbf{r},\tau,\tau'}(u), p^{\mathbf{r}''}_{\mathbf{r},\tau,\tau'}(v))\} = \sum_{r,\tau,\tau'} M_d(p^{\mathbf{r}'}_{r,\tau,\tau'}(u), p^{\mathbf{r}''}_{r,\tau,\tau'}(v)) \cdot p^{r,\tau,\tau'}(u). \quad (6)$$

This is simply the average, for each time and location, of the EMD between the distributions of the next location of u and v. So, for each current location (and time), we use the EMD to compute the dissimilarity between the distributions representing the next locations of users u and v, respectively. The current location is taken according to user u's mobility profile, making this definition asymmetric.

For a particular distance function $d(\cdot)$, the Mallows distance definition can be expanded and previous expressions can be further simplified. This is the case for $d(i,j) := \mathbb{1}_{i\neq j}$, for which $M_d(p,q)$, for arbitrary probability distributions p and q, has closed form $1 - \sum_i \min\{p_i, q_j\}$.

Using the dissimilarity metric, we can compute the *geographic similarity* between the mobility profiles $\langle p(u), \pi(u)\rangle$ and $\langle p(v), \pi(v)\rangle$, for any distance function (e.g., hamming distance, Euclidean distance). For example, considering hamming distance $d(r,r') = \mathbb{1}_{r\neq r'}$, the geographic similarity is:

$$1 - \sum_{r,\tau,\tau'} \left(1 - \sum_{r'} \min\{p^{r'}_{r,\tau,\tau'}(u), p^{r'}_{r,\tau,\tau'}(v)\}\right) \cdot p^{r,\tau,\tau'}(u) = \sum_{r,r',\tau,\tau'} p^{\tau'}_{r,\tau}(u)\pi^{r,\tau}(u) \min\{p^{r'}_{r,\tau,\tau'}(u), p^{r'}_{r,\tau,\tau'}(v)\}. \quad (7)$$

We emphasize that this definition leads to an asymmetrical similarity measure, i.e. the similarity of u to v need not be the same as the similarity of v to u. In principle, this metric can also be computed using measures other than EMD. For example, one can use Kullback-Leibler divergence measure [7] to compute the difference between two probability distributions, ignoring the distance between the locations. We emphasize that we use EMD, in our geographic similarity metric, as we also want to include the distance function $d(\cdot)$ between locations while computing the difference between two distributions (i.e., mobility models).

Consider now the computation of the geographic similarity. For the case, $d(r,r') = \mathbb{1}_{r\neq r'}$, the computation according to closed-form of (7) takes $O(T^2 \cdot R^2)$ operations, where T is the number of time periods and R is the number of locations (regions). For arbitrary $d(\cdot)$ with no closed-form expressions, the geographic similarity is obtained through $T^2 \cdot R$ EMD computations. Each of these EMD computations involves minimizing the Mallows distance, that is equivalent to solving the linear program given by (2).

3.4 Semantic Similarity Metric

The semantic similarity metric builds upon the basic assumption that for two individuals u and v there exists an (unknown) semantics mapping σ of locations $\mathcal{R}$ onto itself (i.e. a permutation) such that $\mathcal{R}$, for u and $\sigma^{-1}(\mathcal{R})$, for v semantically match. It is important to note that assuming such a mapping does not commit us to trying to learn it based on modeling location semantics directly. Instead, we define the (hidden) semantic similarity between u and v as the maximum geographic similarity taken over all possible mappings σ. We define semantic similarity metric as follows.

Definition 3. *Let σ be the mapping of locations of u to locations of v. Let $\mathbf{r}$, $\mathbf{r}'$, and $\mathbf{r}''$ be random variables for locations, and τ and τ' be two time periods. We define the semantic* dissimilarity *between u and v for moving in the sequence of time periods $\{\tau,\tau'\}$ as*

$$\mathcal{D}^v_u(\{\tau,\tau'\}) := \min_{\sigma} \mathbb{E}\left[M_d(p^{\mathbf{r}'}_{\mathbf{r},\tau,\tau'}(u), p^{\sigma(\mathbf{r}'')}_{\sigma(\mathbf{r}),\tau,\tau'}(v))\right], \quad (8)$$

where the Mallows distance $M_d(\cdot)$ is computed over the random variable $\mathbf{r}'$ and the expectation is computed over the random variable $\mathbf{r}$ given time periods τ and τ'.

Now, we define the semantic similarity *between u and v over any sequences of time periods as*

$$\mathsf{sim}_{\mathsf{S}}(u,v) := 1 - \mathbb{E}\left[\mathcal{D}^v_u(\{\tau,\tau'\})\right]. \quad (9)$$

What we compute in (8) is the minimum geographic mobility dissimilarity between u and v where the locations of v are relabeled and mapped to locations of u according to the permutation function σ^v_u (which is the σ that minimizes 8). The intuition is the following. Consider two individuals u and v are at $\mathbf{r}$ and $\sigma(\mathbf{r})$, respectively, at time period τ. The Mallows distance M_d computes how dissimilar their movement will be to the next location which are represented with random variables $\mathbf{r}'$ for u and $\sigma(\mathbf{r}'')$ for v. If, according to a mapping, the way that they move between these locations is similar, they behave similarly with respect to those locations. If this is true across different time periods and location pairs, their mobilities are similar. So, the semantic similarity between two individuals is determined by σ^v_u.

We compute this metric at two different levels of accuracy of the mobility model. If we only consider the visiting probability π part of each individual's mobility profile, we compute $\mathbf{sim_S}$ as follows. Let us consider the hamming distance function $d(r,r') = 1_{r\neq\sigma^{-1}(r')}$. In this case, we can compute the semantic similarity metric as

$$1 - \sum_{\tau} \mathbb{P}\mathrm{r}\{\tau\} \max_{\sigma} \sum_{r} \min\{\pi^r_\tau(u), \pi^{\sigma(r)}_\tau(v)\}. \quad (10)$$

Note that the computation of (10) requires finding the mapping σ which maximizes the inner term for each time period τ. Since there are $R!$ possible candidates for the maximum mapping σ, a brute-force approach is inefficient. However, the problem's structure resembles that of a linear assignment. Focusing on the inner sum, we see that each term (each r) can be associated with R values of $\sigma(r)$ independently of the other components of σ. To recast the problem as a linear assignment, we construct a bipartite graph where the nodes represent $\mathcal{R}$ and $\sigma(\mathcal{R})$, and each edge represents the association (through σ) of r with $\sigma(r)$. The maximum weight assignment of the constructed bipartite graph gives the permutation σ. The running time of this procedure is $O(T \cdot R^3)$ using the Hungarian algorithm [20].

We compute the semantic similarity for the case where we consider the more accurate mobility profile $\langle p, \pi\rangle$ as follows.

$$1 - \sum_{\tau,\tau'} \max_{\sigma} \sum_{r,r'} \pi^{r,\tau}(u) p^{\tau'}_{r,\tau}(u) \min\{p^{r'}_{r,\tau,\tau'}(u), p^{\sigma(r')}_{\sigma(r),\tau,\tau'}(u)\} \quad (11)$$

It is not known whether there is an efficient algorithm to compute the semantic similarity according to (11). The difficulty comes from having to consider assignments of pairs: (r, r') to $(\sigma(r), \sigma(r'))$, which makes this computation resemble the Quadratic Assignment Problem (QAP) [21], known to be NP-Hard and APX-Hard. The semantic similarity (11) can nevertheless be computed through approximation techniques such as Simulated Annealing [5], or the Metropolis-Hastings algorithm [19]. Nevertheless, (11) can be approximated using techniques such as Simulated Annealing [5], or the Metropolis-Hastings algorithm. [19]. We use this algorithm to compute the semantic similarity metric, in the case of considering both visiting and transition probabilities of the individuals' mobility models (see [18] for details). The idea is to find good approximations to the quantity of interest (for us σ) through probabilistic local exploration of the solution space. At each step, we replace our current permutation with a new solution randomly selected from its neighbors (e.g. a permutation which differs in two positions). The output of the algorithm is the best permutation found so far when the algorithm terminates (after some fixed number of iterations). It is known that the starting permutation can have an impact on the quality of the output. In our case, we expect the permutation found during the computation of (10) to be a good starting point.

4. GENERATING FAKE TRACES

In this section, we present the details of our algorithms for sampling fake traces. Figure 2 presents a high-level view. The process of generating and using fake traces are completely separate. When a set of fake traces are generated, they can be used in any protection mechanism accordingly.

4.1 Transform Traces into Semantic Domain

We assume that we have a dataset $\mathcal{S}$ of real traces that we use as seed to generate fake traces. Each seed trace in the dataset comes from a different individual. Generating a fake trace starts by transforming a real trace (taken as seed) to a semantic trace. To this end, we require to know the semantic coordinates of the seed trace. We compute the semantic similarity between all locations in $\mathcal{R}$, and create a location semantic graph $G\langle \mathcal{R}, E, w\rangle$ such that the vertices are in $\mathcal{R}$ and the weight $w_G(r, r')$ on the edge between locations r and r' is the weighted sum of the number pairs of users u and v for whom r and r' is semantically mapped (i.e., $r = \sigma_u^v(r')$), weighted according to their similarity. Then, we create the equivalent semantic classes $\mathcal{C}$ by running a clustering algorithm on this graph. For this purpose, we make use of k-means clustering algorithm, and we choose the number of clusters such that it optimizes the clustering objective. We present the sketch of this algorithm in Figure 2-SemanticClustering().

We then convert the seed location trace *seed* into its corresponding semantic trace *semseed* by simply replacing each location in the trace with all its semantically equivalent locations (according to the semantic classes $\mathcal{C}$). Figure 3 depicts an example of such a semantic seed. Intuitively, this composite trace encompasses all possible geographic traces that have a high semantic similarity to the original seed trace. To be more flexible with respect to the traces that we can generate, we add some randomness to the semantic seed trace. In the transformation process of the seed trace into the semantic trace, we sub-sample locations from the semantic classes as opposed to using them all. For privacy reasons, we remove each location in a cluster with probability par_c. The result is a new cluster $\mathcal{C}'$. We also allow locations of different classes to merge into each other closer to the time instants where the user moves from one class to the other. We implement this by merging a location between two cluster visited Δt time instants away with a geometric probability ${par_m}^{\Delta t}$.

SemanticSimilarity(u, v)

Compute mobility models $\langle p(u), \pi(u)\rangle$ and $\langle p(v), \pi(v)\rangle$

Compute optimal mapping σ_u^v from (8)

Compute semantic similarity $\mathsf{sim_S}(u, v)$ from (9)

Return $\mathsf{sim_S}(u, v), \sigma_u^v$

SemanticClustering$(\mathcal{R}, \mathcal{S}, \kappa)$

Initiate weighted graph G with locations $\mathcal{R}$ as vertices

Forall $u, v \in \mathcal{S}, u \neq v$:

  Set $s_u^v, \sigma_u^v \leftarrow$ SemanticSimilarity(u, v)

  Forall locations $r, r' \in \mathcal{R}$ such that $r' = \sigma_u^v(r)$:

    Set edge weight $w_G(r, r') \leftarrow w_G(r, r') + s_u^v$

Set $\mathcal{C} \leftarrow$ K-Means(G, κ)

Return $\mathcal{C}$

PrivacyTest$(fake, seed, par_s, par_i, par_d)$

Set geographic similarity $sim \leftarrow \mathsf{sim_G}(fake, seed)$

Set intersection between fake and seed $int \leftarrow$ Intersection$(fake, seed)$

Set $dp \leftarrow TRUE$ if
  $\exists s'_{1,\cdots,k-1}$ such that $|\mathsf{sim_S}(s, f) - \mathsf{sim_S}(s', f)| < par_d$

Return TRUE if $int < par_i$ and $sim < par_s$ and dp

GenerateFake$(\mathcal{R}, \mathcal{S}, \kappa, pars)$

Compute aggregate mobility model $\langle \bar{p}, \bar{\pi}\rangle$ by averaging $\langle p(u), \pi(u)\rangle$ over all $u \in \mathcal{S}$

Set $\mathcal{C} \leftarrow$ SemanticClustering$(\mathcal{R}, \mathcal{S}, \kappa)$

Forall $seed \in \mathcal{S}$:

  Set $\mathcal{C}' \leftarrow \mathcal{C}$

  Update $\mathcal{C}'$ by removing each location in any partition with probability par_c.

  Set semantic seed $semseed \leftarrow seed$

  Update $semseed$ by replacing all locations r in the trace with the set of locations in their corresponding clusters, i.e., replace r with ${\mathcal{C}'}_i$ where $r \in {\mathcal{C}'}_i$

  Update $semseed$ by removing the location $r = seed(t)$ at any time t with probability par_l

  Update $semseed$ by merging locations that are located with time distance Δt with probability ${par_m}^{\Delta t}$

  Set $fake \leftarrow$ HMMDecode$(semseed, \langle \bar{p}, \bar{\pi}\rangle)$

  If PrivacyTest$(fake, seed, par_s, par_i, par_d)$

    Set $\mathcal{F} \leftarrow \mathcal{F} \cup fake$

Return $\mathcal{F}$

Figure 2: Fake traces generation algorithm. We present it simplified for the case with a single time period.

4.2 Sample a Trace from the Semantic Domain

Any random walk on the semantic seed trace that crosses the available locations at each time instant is a valid location trace that is semantically similar to the seed trace. However, the synthetic traces we want to generate also need to be geographically consistent with the general mobility of people in the considered area.

We cast the problem of sampling such traces as a decoding problem in Hidden Markov Models (HMMs) [22]. The symbols are locations, the observables are the semantic classes (or the set of semantically equivalent locations in the same class), and the transition probability matrix is our aggregate mobility model. We construct the aggregate mobility model by averaging over the mobility models of all traces in dataset $\mathcal{S}$, as well as giving a small probability to the possible movements between locations according to their distance and connectivity. More precisely, we compute the aggregate transition probability $\bar{p}_r^{r'}$ as $\frac{\sum_{u \in \mathcal{S}} p_r^{r'}(u) + \epsilon \cdot max(1, d(r,r'))^{-2}}{z_r}$, where ϵ is a small constant, and z_r is the normalizing factor. We compute the aggregate visiting probability $\bar{\pi}^r$ as the average of $\pi^r(u)$, for all $u \in \mathcal{S}$. The probability distribution $\bar{\pi}$ is also the steady state probability distribution of $\bar{p}$.

By decoding the semantic trace into geographic traces using HMMs, we generate traces that are probable according to aggregate mobility models, i.e., there could be one individual who prefers to take that trajectory. There are different HMM decoding algorithms. We make use of the Viterbi algorithm which is a dynamic programming algorithm to generate the most probable trace given the observation (i.e., the sematic seed trace) [30]. More precisely, Viterbi finds

$$\arg\max_{fake} Pr\{fake | semseed(t), \langle \bar{p}, \bar{\pi} \rangle\}$$

assuming that $fake(t)$ can only choose from locations in $semseed(t)$. Finding the most likely fake trace is equivalent to finding the shortest path in an edge-weighted directed graph where each location at time instant t is linked to all locations at the subsequent time in the semantic seed trace.

By using this encoding technique, we make sure that the sampled trace is consistent with the generic mobility and has a significant probability of (geographically) being a real trace. However, Viterbi produces (only) one trace, hence we cannot directly generate multiple fake traces. To address this issue, we add randomness to the trace reconstruction of Viterbi. We modify the Viterbi algorithm, which originally, at each step (time instant) selects the most probable location in the path; we add some randomness to the probabilities such that the algorithm does not deterministically select the most probable location. More precisely, we slightly perturb the probabilities in such a way that Viterbi selects randomly among a set of locations that are close in probability to the most probable location. We implement this idea by choosing a parameter par_v and multiplying all the probabilities of moving from one location to the next with a random number between 1 and par_v.

4.3 Threat Model

The threat against fake trace generating algorithms is twofold. (1) One threat is directly related to the adversary who wants to filter out traces that are fake and to find out the true location of mobile users (*localization* attack), e.g., when it is used in location based services to hide the true location of the user. For this attack, we assume the adversary has a background knowledge on general mobility models of users in the considered area. In Section 5.6, we quantify the success rate of an adversary in localizing users while fake traces are used as a defense mechanism. (2) The other threat is secondary as it is related to the algorithm that generates the fakes, and comes from the adversary whose goal is to identify the real individuals whose trace are used as seed (*membership inclusion* attack). In the next subsection, we enforce a privacy property that protects against the second threat. We guarantee plausible deniability for seed contributors independently from the adversary's knowledge.

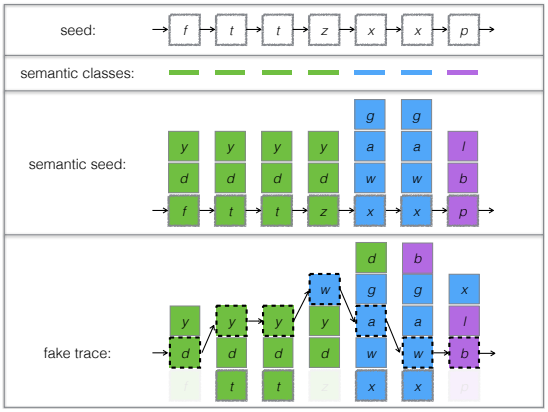


Figure 3: A sketch of generating a fake trace from a seed. Each location is represented by an English letter in a box. The semantic class associated with each location is represented by a different color. The semantic seed trace is a trace that includes the locations in the seed along with other locations in the same cluster at each time instant. Here, locations are clustered as $\{y, d, f, t, z\}, \{g, a, w, x\}, \{l, b, p\}$. To generate a fake trace, we first probabilistically remove the seed location and probabilistically merge subsequent classes. In our example, f, z, p are removed, and w, d, b, x are merged into their neighboring visited clusters. We then run a decoder to generate a probable trace given the possibility of choosing from all available locations at each time instant. The fake trace is shown with dashed boxes. A rejection test will run on this trace to guarantee its privacy compliance.

4.4 Privacy Tests

In the end, we want to make sure that the generated fake traces do not leak information about the seed trace. We design tests to protect against the following threat model. We assume the adversary has access to traces (or mobility patterns) of some individuals that might overlap with the set of individuals who contribute to the seed dataset. The attacks depend on the scenario in which a fake trace is used. The adversary might be interested in separating fake from real (in LBS scenario) or finding the seed from which a fake trace is generated (in trace publishing scenario). To protect against such threats, as the last step of our process, we run a PrivacyTest() on each of the generated fake traces.

– We compute the geographic similarity of each fake trace to the seed trace and reject the fake trace if its similarity is higher than a threshold par_s. This makes sure that the fake trace does not *statistically* leak information about the mobility of the individual behind the seed trace. We also reject a fake trace if its intersection with the seed trace is

larger than par_i. This makes sure that the exact locations visited by the individual are not present in the fake trace. These tests provide the privacy guarantee with respect to information leakage of visited locations.

– We also use another notion of privacy, which is more of relevance in the case of publishing a location dataset. Specifically, we want to defend against *membership inclusion* attacks, in which an adversary wants to infer whether a particular individual's data was included in the seed dataset. Therefore, we design another privacy test that guarantees *plausible deniability* for those whose trace was used as seed. The main idea is that a fake trace should be as semantically similar to its own seed as to some other real traces which are not included in the seed dataset, so that an adversary cannot certainly infer that a particular individual was in the seed dataset and de-anonymize the seed contributor. Intuitively, if the semantic similarity of a fake trace to its own seed is comparable to its semantic similarity with some non-seed real trace, then the fake trace could have been generated from either of them. To enforce this property, we test if for a generated fake trace f (that was generated from a seed trace s), there are some other real trace s' such that their differential semantic similarity is bounded.

$$|\mathsf{sim}_\mathsf{S}(s, f) - \mathsf{sim}_\mathsf{S}(s', f)| < par_d$$

We can enforce this to hold for a minimum $k-1$ number of alternative s' traces (from which we are not going to publish fake traces). Thus, there is at least one real trace outside the set of seeds associated with fake traces that could have produced each releasable fake trace. More generally, we enforce the size of anonymity set to be k. This property, which provides *plausible deniability*, is conceptually related to, but weaker than, Differential Privacy (DP) [9]. Indeed, this is one kind of guarantee that differential privacy is meant to provide.[3] That said, we enforce this property for *actual* seeds in the datasets. Thus, by looking at a fake trace the adversary cannot surely conclude that a particular trace was in the seed dataset, because there exist at least $k-1$ other traces that could have seeded the same fake trace.

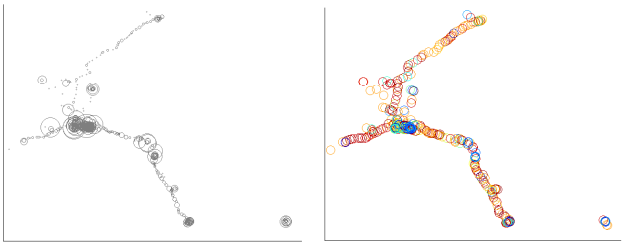
Each time we generate a new fake trace that passes the privacy tests, we compute its likelihood based on the aggregate mobility model. One can then randomly sample from the bag of fake traces based on this likelihood. The traces that are generated according to this process do not leak information about the seed traces, yet they share their average geographic features and semantic features.

4.5 Discussion

The fake generation process, which results in a pool of fake traces having passed the privacy test, is run offline on powerful machines, before the users' device retrieve and use such fakes. Therefore, this computational burden is not placed on the users' device. Nevertheless, the generation process is reasonably efficient: the computation of both the aggregate mobility and the semantic clustering needs to be done only once for each input set of real traces. The former takes time $O(SL+(RT)^2)$ where $S = |\mathcal{S}|$ is the number of seed traces, L is the length (i.e., number of events) of each seed trace. The latter is dominated by $S(S-1)$ semantic similarity computations (e.g., each taking $O(TR^3)$ in the zeroth-order case) and one clustering operation. Excluding the final clustering, this step is embarrassingly parallel: the semantic similarity for any two users u, v can be computed independently. Also, if a few input traces are added, both the aggregate statistics and the semantic clustering can be updated and do not need to be recomputed from scratch. Once the semantic clustering has been computed, an arbitrarily large number of fakes for each seed can be generated. This process is also embarrassingly parallel, since each fake can be generated independently of other fakes for that seed, and other seeds.

The algorithm of Figure 2 works if the input dataset contains at least two seed traces. However, its quality will be high if, among the seed traces: the coverage of the location set $\mathcal{R}$ is high (not necessarily complete); the semantic similarity is high; and the geographic similarity is low.

[3]DP can be thought of as providing plausible deniability by thinking of the differing elements between the two neighboring datasets as two possible inputs of the same user. When DP is satisfied, the output distribution is approximately the same, so that that user can plausibly pretend having used any one of the two inputs.



(a) Visited Locations. The size of locations are proportional to their total visits.

(b) Visited locations colored according to their semantic clustering (20 clusters).

Figure 4: 400 locations visited around Lausanne and nearby towns by the 30 users. Some users commute between two towns whereas the majority of them live and work in the same city of Lausanne (the area with higher concentration).

5. EVALUATION

In this section, we run our algorithms on a set of real location traces and evaluate their utility and privacy in two scenarios: publishing a location dataset, and sharing locations with a location-based service.

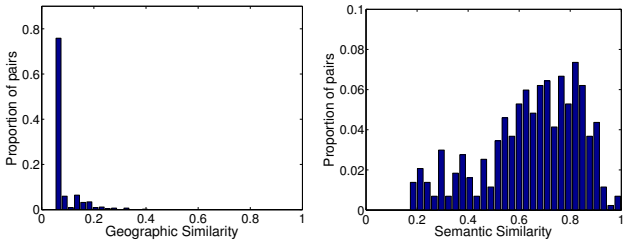
5.1 Dataset

The dataset we use for the evaluation is collected through the Nokia Lausanne Data Collection Campaign[4] (see [14]). We prepare the dataset for our needs in two phases, filling gaps in the traces and discretizing the time and location.

The raw dataset contains events of three types: GPS (the GPS position of the user is known), WLAN (the SSID and signal strength of a set wireless networks which surround the user are known), and GSM (the identifier of the GSM base station to which the user's phone is associated is known). In the first phase, we compute valid traces (out of possibly partial traces) by aggregating events and filling gaps. We do this by interpolating along the path of consecutive GPS points and using the WLAN and GSM information.

In the second phase, we extract two days of traces for 30 users, such that each trace contains a sequence of 72 locations (i.e., one location is reported every 20 minutes). Some locations are visited very rarely only by very few users.

[4]http://research.nokia.com/page/11367



(a) (b)

Figure 5: Normalized histogram of the (a) geographic similarity and (b) semantic similarity of all distinct pairs of 30 users. (a) Mobility models of different individuals is geographically very specific to themselves, i.e., they are unique. This is well reflected in the skewed distribution of geographic similarity towards very small values. (b) As hypothesized in this paper, majority of individuals have high semantic similarities with respect to their mobility models.

Thus, we reduce the number of locations from 1491 to 400 by clustering close-by locations together. We use a hierarchical clustering algorithm for this purpose, in which the distance between two locations is taken to be proportional to both the Euclidean distance between the locations and the product of their weights (defined as the number of total visits to each location, for all users). This means that locations clustered together will tend be both geographical close and have few visits. The geographical distribution of visits of all users over the locations in the considered area is shown in Figure 4(a).

We computed the mobility profiles of all 30 users, and then the semantic location graph by calculating a similarity score for each pair of locations, averaged across all users. After clustering this semantic location graph, we obtained 20 location clusters. We choose this number of clusters as it provides optimal clustering i.e., it maximizes the ratio of inter-cluster similarity over intra-cluster similarity. This clustering is illustrated in Figure 4(b), where each location is drawn with the color of the cluster it belongs to. The figure allows us to distinguish some patterns, for example locations at the center of cities are mostly in blue, while many locations representing roads and highways are colored in red. Also notice that the semantic clustering does not seem to depend on the geographical distance of locations.

To illustrate the difference between geographic and semantic similarities, we can compute those metrics pairwise over all 30 users.[5] The result is shown in Figures 5(a), and 5(b). The first histogram shows that the 30 users are not strongly geographically similar to each other, except for a few pairs of users. This is expected given the range of locations they explore overall, as seen in Figure 4(a). On the other hand, the distribution of the semantic similarity across all distinct pairs of users has a larger variance; while some pairs of users are not similar at all (e.g., those with semantic similarity score of 0.2), a large number of users are highly similar.

5.2 Fake Trace Generator Tool

We build our tool to generate fake traces on top of the open-source Location Privacy Meter (LPM) [27]. To exploit LPM's modularity we split our algorithm into modules. To implement the time-dependent sub-sampling of clusters and merging around transitions, and the transformation of users' actual traces into semantic traces, we use the location obfuscation mechanism feature of the tool. The reconstruction of geographically valid synthetic traces from the semantic traces is done using the Viterbi algorithm (implemented in LPM as a tracking attack). To cluster the semantic location graph, we employ the CLUTO toolkit [31].

[5]Since both the geometric similarity, and the semantic similarity of a user with herself is 1.0, we exclude such pairs.

5.3 Simulation Setup

As for the parameters of the GenerateFake() algorithm, we set the location-removal probability par_c to 0.25, and we set the location merging probability par_m to 0.75. We set the probability par_l of removing the true location visited in the seed to 1.0. We set the randomization multiplication factor for Viterbi randomization to 4. So, for each probability assigned to each location at each time instant, we multiply it with a randomly chosen number between 1 to 4. We set very tight values for the privacy parameters. We set par_i, the maximum intersection between fake and seed, to 0. So, we do not tolerate any intersection between fake and seed. We set the geographic similarity threshold par_s to 0.1, and the differential semantic similarity threshold also to 0.1.

For each of the 30 users, we generated about 500 fake traces. Out of those we randomly pick 50 traces (for each user) to be used for the datasets publishing scenario. For the LBS scenario, we sampled traces (for each user) according to the traces likelihoods, out of the pool of traces (for that user) which passed the privacy test.

Out of the two days of traces (each 72 timestamps, for each of the 30 users), we use the first day as the training dataset, and the second day as the testing dataset. We calculate the aggregate statistics and mobility profile of users on the training dataset, while we use the testing dataset to evaluate both the data publishing scenario and the attack for the LBS scenario. Unless otherwise stated, for all experiments, we consider a single time-period, and compute the zeroth order versions of the geographic and semantic similarities.

5.4 Evaluation Metrics

In the following two subsections we evaluate the use of fake traces in two popular scenarios: publishing a dataset of fake traces, and using fake locations along with real locations when accessing location-based services. In both scenarios, we evaluate our fake traces with respect to two metrics: *privacy* and *utility*. The privacy guarantee that we provide using our privacy rejection test applies to both scenarios. However, there are some differences in terms of the adversary model between different scenarios. There are therefore some additional considerations regarding the privacy of users in location-based services, e.g., their privacy against inference attacks, that we discuss in its corresponding subsection. The utility metric is very dependent on the application (scenario), hence is measured differently in each case.

5.5 Scenario: Publishing Location Datasets

5.5.1 Setup

In this scenario, we assume that we generate a fake trace for some seed traces and publish them all in a dataset. We use some real traces in our dataset that we use as alternative seeds in the differential semantic similarity test.

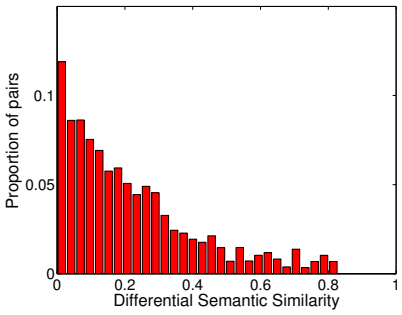


Figure 6: Histogram of the differential semantic similarity between fake and real traces. It presents the distribution of the absolute difference $|\mathsf{sim}_S(s, f) - \mathsf{sim}_S(s', f)|$, for all pairs of f (plus its seed s) and s'.

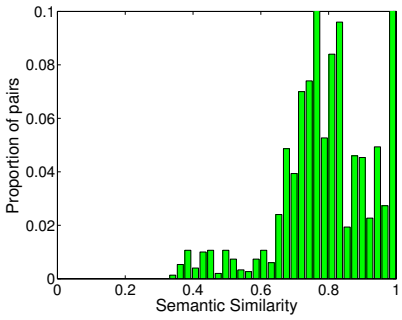


Figure 7: Normalized histogram of the semantic similarity of all distinct pairs of: each of the 30 real traces, along with their associated fake traces.

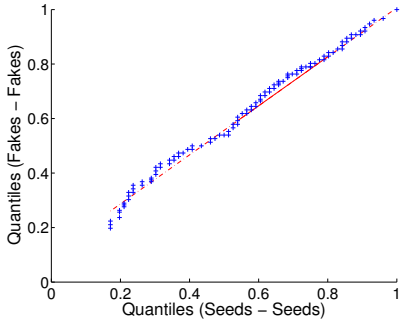


Figure 8: Q-Q plot for comparing two distributions: semantic similarity among all real seed traces, and semantic similarity among all fake traces. The plot shows a very strong correlation between two distributions.

5.5.2 *Privacy*

We assume an adversary wants to infer information about the true traces that have been used to generate the fakes. Due to the privacy test, location privacy of individuals who contributed to the seed dataset is guaranteed as fake traces do not intersect with the locations that are visited and does not even leak statistically about what could have been visited by those individuals. Moreover, due to the differential semantic similarity test, the seed traces are not the only traces that could have generated the released fake traces.

Out of all fake traces that we generated from our 30 seed traces, on average 80% of them could pass the geographic and intersection privacy tests with tight constraints ($par_i = 0$, i.e., no intersection allowed, and $par_s = 0.1$), so it is not difficult to generate fake traces that satisfy such privacy guarantees. Regarding the differential semantic similarity test, we should be able to find enough number of real traces as alternative traces that could have been the seed for releasing fake traces. In Figure 6, we show the difference between semantic similarity of a fake trace and its seed with the semantic similarity of the same fake trace and any other real trace in our dataset. The histogram shows that the majority of fake traces have very low similarity to real traces other than their seeds. This is due to the high semantic similarity between real traces (Figure 5b) so it is not difficult to find potential alternative seeds for a fake trace. We set par_d to 0.1 to obtain a high level of differential semantic privacy.

5.5.3 *Utility*

To preserve the utility of the original traces, fake traces should share similar statistical properties with the real trace dataset. Note, however, that we would not expect all useful statistics to be preserved since some may be counter to our goal of preserving privacy. That is, certain geographic features are expected not to be preserved, due to the nature of the generation and the privacy test. For example, if a location is primarily visited by a single user in the real dataset, it is unlikely that the location would be visited with similar frequency by a (fake) user in the fake trace dataset. This is because if such a fake trace were generated from that seed, the privacy test would reject it. Nevertheless, we can evaluate to what extent certain useful statistics are preserved.

To start, we compare the basic mobility statistics obtained from the real and fake datasets. We compute the aggregate mobility model for each fake dataset (we generate 10 of size 30), and compare its geographic similarity with the real dataset. More precisely, for a fake dataset $\mathcal{F}$, we compute $\langle \bar{p}_\mathcal{F}, \bar{\pi}_\mathcal{F} \rangle$ and compute its similarity to $\langle \bar{p}, \bar{\pi} \rangle$. The statistical similarity of $\bar{p}_\mathcal{F}$ with $\bar{p}$ over all fake datasets is

$$[0.8061, \quad 0.8073, \quad 0.0060]$$

on (average, median, standard deviation), and the results for the statistical similarity of $\bar{\pi}_\mathcal{F}$ with $\bar{\pi}$ is

$$[0.7856, \quad 0.7867, \quad 0.0152].$$

Both these results show a strong correlation between average/aggregate mobility information of real and fake datasets.

We then compare the location visiting probabilities of the real dataset and fake datasets. Namely, for each dataset we compute the spatial allocation, i.e., for each location (from least to most popular, for that dataset), we calculate the number of visits spent in that location across all traces in the dataset. We then normalize this quantity to obtain a probability distribution over locations (sorted by popularity), i.e., for each location we have the probability of a random visit to that location. From these distributions, we compute the KL-divergence of the real (training) dataset to each of our fake datasets, and to a variety of baselines.[6] The results are shown in Table 2. Since the KL-divergence is not upper bounded, we use as baselines the KL-divergences of the real (training) dataset to the following distributions: real testing dataset; uniform visiting probabilities; and single location visiting. We see that while the KL-divergence of the real (training) dataset to the real testing dataset is smaller than that to the fake datasets, the latter is also significantly smaller than both the the KL-divergences to the uniform visiting baseline and the single location visiting baseline. This indicate that a lot of information is preserved in the fake datasets. Next, we repeat the previous calculation of KL-divergence, but considering only visits to the 50 most popular locations (of each dataset). Table 3 shows the re-

[6] Because the KL-divergence is only defined at points where the second distribution is not zero, unless the first is also zero, we set all zero probabilities to $\epsilon = 0.1$, before normalizing. This is required because there may be locations which are visited in the fake dataset but not in the real dataset, or vice-versa.

sults: the information is almost as well preserved in the fake datasets than compared to the real testing dataset.

We also compare the users time allocation of the real and fake datasets. Namely, for each dataset and each user, we calculate the time spent at each location, among the locations visited. That is, we calculate, for the three most popular locations of that user, what proportion of the time is spent in each. We perform this calculation across all 30 users and normalize the result. We compare this distribution for the real and fake datasets. Table 4 shows the KL-divergence of the real (training) dataset to the fake datasets and baselines: real testing dataset; uniform time allocation (each user spends $1/k$ proportion of time at each of the k locations); random time allocation (each user spends a uniformly random proportion of time at the location). This statistic is highly preserved in the fakes; sometimes the fake datasets' distribution is closer to that of the real (training) dataset, than the distribution of the real testing dataset is.

The previous results provide confidence that useful information is indeed preserved in the fake traces dataset. That said, our original goal was to preserve utility in the sense of semantic similarity, so it sensible to wonder how close we are to that goal. To determine this, we first compute the semantic similarity of each fake trace with its own seed trace to check if the semantic features of the original traces are indeed preserved. Figure 7 illustrates the distribution of this value over all fake traces. Clearly, the distribution is biased towards higher similarity values. So, the fake traces considerably preserve the semantic features of the real traces.

Another type of statistics that we would expect the set of fake traces to preserve is the inner similarity between the set of traces. In Figure 8, we present the correlation between two distributions: semantic similarity among real traces, and semantic similarity among fake traces. The Q-Q plot shows a significant correlation between these two distributions; they are strongly linearly related. This reflect that in addition to maintaining the information about each seed, we also preserve the statistical relation among the traces.

Results show that fake traces cannot be distinguished from the real ones if it appears among some real traces. This is because the relation between a fake trace and the distribution of real traces is largely indifferent from that of a real trace with respect to both semantic and geographic features.

5.6 Scenario: Using Location-based Services

The utility and privacy evaluation for the publishing dataset scenario applies to the case where traces are shared with a service provider. However, we can perform a more specific analysis on the fake locations when they are shared in a new setting. We present the details of how fake locations are used to protect location privacy, and how they perform against inference attacks despite the fact that they have passed privacy guarantee tests.

Testing	Fakes		Uniform	Single
0.0377	Mean	Std	1.1918	4.6666
	0.3841	0.0432		

Table 2: KL-divergence of the location visiting probabilities of the real (training) datasets against the 10 fake datasets, and various baselines. "Testing" is the testing portion of the real dataset (see Section 5.3); "Uniform" is the uniform distribution over all locations; and "Single" is the distribution where all users always visit the same location.

5.6.1 Setup

We assume a user shares her current location with a location-based service with a probability β (set to 0.5 in our case). The service provider, in return, provides the user with contextual information about the shared locations (e.g., list of nearby restaurants, current traffic information on the road). The service provider would receive a sequence of locations that are visited by the user at different time instants. To protect her location privacy, i.e., hiding her location at the time of access to the LBS and also preventing the inference of the full trajectory, the user sends a number of fake locations along with her true location. We assume the user has access to some full fake traces, and at any time instant t, when she is accessing the server, she consistently adds the locations visited at t on each fake trace to her actual location at t and sends them to the server. The service provider responds to each of these location queries, and the user needs to filter out the results associated with fake locations to obtain the information about her true location.

We evaluate a few other methods to generate fake locations along with our method for comparison.

- Uniform IID: We sample each fake location independently and identically distributed from the uniform probability distribution.
- Aggregate Mobility IID: We sample each fake location independently and identically distributed from the aggregate mobility probability distribution $\bar{\pi}$.
- Random Walk on Aggregate Mobility: We sample a fake trace by doing a random walk on the set of locations following the probability distribution $\bar{p}$.
- Random Walk on User's Mobility: We do a random walk on the set of locations following the probability distribution $p(u)$ to generate a fake trace.

5.6.2 Privacy

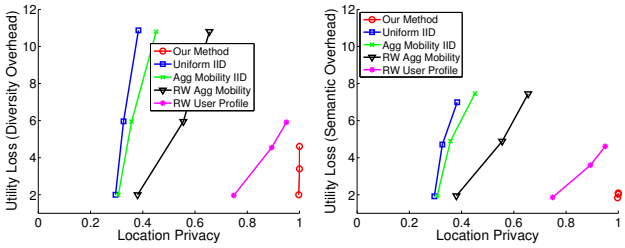
We assume the adversary wants to filter out the fake locations and to find the true sequence of locations that are

Testing	Fakes		Uniform	Single
0.0215	Mean	Std	0.2040	5.1131
	0.0289	0.0086		

Table 3: KL-divergence of the 50 most popular location visiting probabilities of the real (training) datasets against the 10 fake datasets, and various baselines (see Table 2).

	Testing	Fakes		Uniform	Random
1st	0.0189	Mean	Std	0.1652	0.6794
		0.0125	0.0022		
2nd	0.0026	Mean	Std	0.0778	0.5360
		0.0092	0.0031		
3rd	0.0114	Mean	Std	0.0779	0.5092
		0.0089	0.0036		

Table 4: KL-divergence of the the users time allocation distribution among the three most popular locations (of each user) of the real (training) datasets against the 10 fake datasets, and various baselines.



(a) (b)

Figure 9: Location privacy versus utility loss for different fake generating algorithms. The privacy is measured as probability of error of adversary in guessing the correct location of users. We plot the median location privacy across all 30 users. The probability of connecting to the LBS is set to 0.5, so one half of the time instants the users connect to the server. We evaluate the use of $1, 5, 10$ fake traces, hence three dots for each algorithm. (We repeated the experiment 20 times and took the average: 4 times with a different selection of fake traces, and for each of such selection, 5 times to eliminate the randomness of exposures.) The utility loss is the number of distinct locations that are sent to the server (9a), and the number of (distinct) semantic classes (i.e., clusters) exposed for each event (9b).

visited by the user. The privacy metric that we use is based on the error of adversary in his inference attack [27]. Put simply, the fraction of true locations that are missed by the adversary is our privacy metric. More precisely, the metric is the probability of error of inference attack on guessing the correct location. We assume the adversary makes use of the aggregate mobility model $\langle \bar{p}, \bar{\pi} \rangle$ to single out the true locations and reconstruct the true location of the user.

5.6.3 Utility

There is an overhead to these privacy-preserving mechanisms, as a user has to send more than one query to the server to get the results for one query. This can be interpreted as utility loss, and so we define two metrics for (the lack of) utility. The first is the number of distinct locations sent by the user at each time (diversity overhead). Note that this number can be less than the number of fake traces as they might intersect at the times of connection to the server. Additionally, some service providers (e.g., Google Now) might profile the user over time based on the type of locations she visits, in order to provide recommendations or reminders. In these cases, the queries that are sent to the server can pollute the profile of the user hence reduce the predictability power of the service provider. For this, we use the number of (distinct) semantic clusters among the locations sent by the user at each time (semantic overhead).

5.6.4 Results

Figure 9 shows the tradeoff between location privacy and utility for various methods of generating fake traces. We evaluate the utility loss in terms of two metrics: diversity overhead (Figure 9a), and semantic overhead (Figure 9b). We evaluate the privacy for exposure probability $\beta = 0.5$, and three different number of fake traces: $1, 5, 10$. Although the number of fake traces are the same, across different algorithms, but the average number of distinct locations sent to the LBS is not the same. This is because of the high randomness in the *Uniform IID*, *Agg Mobility IID*, and *RW Agg Mobility* that select fake traces from all possible locations. Our method and the *RW User Profile* method have both lower diversity overhead and lower semantic overhead in the set of fake locations. Our method, clearly outperforms all the tested methods, especially the random strategies. For the case of *RW User Profile* method, the privacy level against tracking attack gets closer to what we achieve (which is very close to the maximum), due to the fact that the fake traces generated by *RW User Profile* are geographically very similar to the true location of the user, and hence creates high confusion, hence error, for the adversary. Note that the *RW User Profile* is never a privacy-preserving fake injection method as the adversary can easily de-anonymize and profile the user, no matter if he makes mistakes on exactly tracking the user at each access time (as shown here).

Whereas, our method is ensured to have minimal geographic mutual information with the true trace, thus it is robust against profiling attack. We also ensure that the fake traces have small differential semantic similarity, thus they are robust against de-anonymization. Additionally, the plot shows that our method is the strongest fake generating algorithm against an attacker who is interested in filtering out the fake locations and localize the user over time.

6. RELATED WORK

Location obfuscation is a prevalent non-cryptographic technique to protect location privacy. It does not require changing the infrastructure, as it can also be done all on the user's side either by altering (perturbing) the location coordinates to be reported or by sending fake location reports interleaved or along with the true location of the user.

Many location perturbation techniques have been proposed in the literature, usually based on adding some noise to the user's location coordinates or reducing its granularity, e.g., [1, 2, 10, 28]. The downside of these techniques is that they reduce the service quality of the user in interaction with the location-based service (LBS) provider. This is because the server provides contextual information related to the shared location and not the true location of the user. So, users have to trade service quality to obtain their required level of privacy. Optimal solutions for location perturbation techniques are proposed [4, 28] which show the high cost of location privacy on service quality using perturbation.

Hiding the user's true location among fake locations is a promising yet very little-explored approach to protecting location privacy. There are few simple techniques proposed so far: adding independently selected fake locations drawn from the population's location distribution [26], generating dummy locations at random as a random walk on a grid [13, 32], constructing fake driving trips by building the path between two random locations on the map given the more probable paths traveled by drivers [15], or adding noise to the paths generated by road trip planner algorithms [6]. These solutions lack a formal model for human mobility and do not consider the semantics associated with sequence of locations visited by people over time. Thus, the generated traces can be distinguished from real location traces.

This paper, to the best of our knowledge, is the first that proposes a systematic methodology for generating fake location traces based on statistical features of both geographical

and semantic dimensions of real traces, and based on a metric to measure how realistic a synthetic trace is. Moreover, we introduce multiple privacy tests to ensure that the published/shared fake traces themselves do not leak information about real seed traces. Our evaluation on real data also shows the clear advantage of our algorithm with respect to other existing approaches against known inference attacks.

7. CONCLUSIONS

This is the first paper to address the problem of generating realistic synthetic location traces based on a quantitative metrics. Generating such traces is very useful to protect privacy of users when they share location with location-based services, or when we want to publish a dataset of locations to be used for various research reasons. Based on well-established statistical methods, we propose two metrics to quantify geographic and semantic features of human mobility. Using these metrics, we propose efficient algorithms to generate fake traces that do not leak geographic information about real individuals (guaranteed using a privacy rejection test), yet highly resemble the mobility of a population semantically. We show that inference attacks cannot identify the true location of mobile users if our fake traces are used as protection. We also quantitatively show that our method is superior to all existing methods of generating fake traces.

8. REFERENCES

[1] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 901–914. ACM, 2013.

[2] C. A. Ardagna, M. Cremonini, S. De Capitani di Vimercati, and P. Samarati. An obfuscation-based approach for protecting location privacy. *Dependable and Secure Computing, IEEE Transactions on*, 8(1):13–27, 2011.

[3] O. Berthold and H. Langos. Dummy traffic against long term intersection attacks. In *Privacy Enhancing Technologies*, pages 110–128. Springer, 2003.

[4] N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi. Optimal geo-indistinguishable mechanisms for location privacy. *CoRR*, abs/1402.5029, 2014.

[5] S. Brooks and B. Morgan. Optimization using simulated annealing. *The Statistician*, pages 241–257, 1995.

[6] R. Chow and P. Golle. Faking contextual data for fun, profit, and privacy. In *WPES '09: Proceedings of the 8th ACM workshop on Privacy in the electronic society*, pages 105–108, New York, NY, USA, 2009. ACM.

[7] T. M. Cover and J. A. Thomas. *Elements of information theory*. John Wiley & Sons, 1994.

[8] C. Diaz and B. Preneel. Taxonomy of mixes and dummy traffic. In *Information Security Management, Education and Privacy*, pages 217–232. Springer, 2004.

[9] C. Dwork. Differential privacy. In M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, editors, *33rd International Colloquium on Automata, Languages and Programming, ICALP 2006*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2006.

[10] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. Preserving privacy in gps traces via uncertainty-aware path cloaking. In *CCS '07: Proceedings of the 14th ACM conference on Computer and communications security*, pages 161–171, New York, NY, USA, 2007. ACM.

[11] D. C. Howe and H. Nissenbaum. TrackMeNot: Resisting surveillance in web search. *Lessons from the Identity Trail: Anonymity, Privacy, and Identity in a Networked Society*, 23:417–436, 2009.

[12] A. Juels and R. L. Rivest. Honeywords: Making password-cracking detectable. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 145–160. ACM, 2013.

[13] H. Kido, Y. Yanagisawa, and T. Satoh. An anonymous communication technique using dummies for location-based services. In *Pervasive Services, 2005. ICPS '05. Proceedings. International Conference on*, pages 88–97, July 2005.

[14] N. Kiukkonen, J. Blom, O. Dousse, D. Gatica-Perez, and J. Laurila. Towards rich mobile phone datasets: Lausanne data collection campaign. *Proc. ICPS, Berlin*, 2010.

[15] J. Krumm. Realistic driving trips for location privacy. In *Pervasive '09: Proceedings of the 7th International Conference on Pervasive Computing*, pages 25–41, Berlin, Heidelberg, 2009. Springer-Verlag.

[16] E. Levina and P. Bickel. The Earth Mover's distance is the Mallows distance: some insights from statistics. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 251 –256 vol.2, 2001.

[17] A. Machanavajjhala, D. Kifer, J. Abowd, J. Gehrke, and L. Vilhuber. Privacy: Theory meets practice on the map. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 277–286. IEEE, 2008.

[18] D. J. MacKay. *Information theory, inference, and learning algorithms*, volume 7. Citeseer, 2003.

[19] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[20] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial & Applied Mathematics*, 5(1):32–38, 1957.

[21] P. M. Pardalos, H. Wolkowicz, et al. *Quadratic Assignment and Related Problems: DIMACS Workshop, May 20-21, 1993*, volume 16. American Mathematical Soc., 1994.

[22] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[23] D. B. Rubin. Statistical disclosure limitation. *Journal of Official Statistics*, 9(2):461–468, 1993.

[24] Y. Rubner, C. Tomasi, and L. Guibas. A metric for distributions with applications to image databases. In *Computer Vision, 1998. Sixth International Conference on*, pages 59 –66, jan 1998.

[25] Y. Rubner, C. Tomasi, and L. J. Guibas. The Earth Mover's Distance as a Metric for Image Retrieval. *International Journal of Computer Vision*, 40:99–121, 2000. 10.1023/A:1026543900054.

[26] R. Shokri, G. Theodorakopoulos, G. Danezis, J.-P. Hubaux, and J.-Y. Le Boudec. Quantifying location privacy: the case of sporadic location exposure. In *Proceedings of the 11th international conference on Privacy enhancing technologies*, PETS'11, pages 57–76, Berlin, Heidelberg, 2011. Springer-Verlag.

[27] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux. Quantifying location privacy. In *Proceedings of the 2011 IEEE Symposium on Security and Privacy*, SP '11, pages 247–262, Washington, DC, USA, 2011. IEEE Computer Society.

[28] R. Shokri, G. Theodorakopoulos, C. Troncoso, J.-P. Hubaux, and J.-Y. Le Boudec. Protecting location privacy: optimal strategy against localization attacks. In T. Yu, G. Danezis, and V. D. Gligor, editors, *ACM Conference on Computer and Communications Security (CCS'12)*, pages 617–627. ACM, 2012.

[29] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, 2010.

[30] A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on*, 13(2):260–269, 1967.

[31] R. W. White, A. Hassan, A. Singla, and E. Horvitz. From devices to people: Attribution of search activity in multi-user settings. In *Proc. International World Wide Web Conference (WWW)*, 2014.

[32] T.-H. You, W.-C. Peng, and W.-C. Lee. Protecting moving trajectories with dummies. In *Mobile Data Management, 2007 International Conference on*, pages 278–282, May 2007.